

AMENDMENTS TO THE SPECIFICATION

Please replace paragraph starting at line 11, page 6, with the following amended paragraph:

FIG. 1 depicts a partial referencing graph for an illustrative mutator thread executing on a processor having memory referenceable as address space 101. The mutator thread is at a current function corresponding to stack frame 111 in a calling hierarchy represented as activation record stack 110. Activation record stack 110 and stack frame 111 are described in greater detail below. A register context 120 is associated with the current function. Register context 120 may be a full set of machine registers available to the mutator thread or a window into a subset of machine registers allocated to the current function. A portion of address space 101 is dynamically allocable as heap 102. Individual memory objects (e.g., objects 150, 151, 131, 132, 133, 134, 135 and 141) are dynamically allocated from heap 102 and, in the referencing graph of **FIG. 1**, are currently reachable by a set of pointers including local variable L1, external or outer variable E1, and contents of registers R3 and R6 of register context 120. Register context 120 may include a portion organized as a stack. In some configurations, a register context and frame for a particular function may be combined in a unified storage arrangement.

Please replace paragraph starting at line 6, page 19, with the following amended paragraph:

A variety of execution environments and mutator code instruction selections have been described herein. Fig. 6 depicts a flowchart illustrating preparation by a compiler of mutator object code and interaction of such code, collector code and suspension trap handler code with root set storage and trapping value storage 690, in accordance with various exemplary embodiments of the present invention. Compiler 610 is illustrative of functions performed by mutator code preparation facilities including traditional batch mode

compiler implementations as well as just-in-time (JIT) compiler implementations, though JIT compiler implementations may omit some functions, such as lexical analysis, and also include a tighter coupling with actual execution of compiled code. Other mutator code preparation facilities, including other compilers, may implement differing sets and/or sequences of operations while still providing mutator code with thread suspension facilities in accordance with the various embodiments described herein.